PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of Docket No: A7870

Rauf IZMAILOV, et al.

Appln. No.: 09/897,495 Group Art Unit: 2662

Confirmation No.: 2079 Examiner: Saba TSEGAYE

Filed: July 03, 2001

For: PATH PROVISIONING FOR SERVICE LEVEL AGREEMENTS IN DIFFERENTIATED SERVICE NETWORKS

DECLARATION UNDER 37 C.F.R. § 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

We, Raul Izmailov, Subir Biswas, and Samrat Ganguly, hereby declare and state as follows:

1. We are the inventors named in the above-captioned U.S. Application No. 09/987,495, filed July 3, 2001, which claims priority to U.S. Application No. 60/243,731 filed October 30, 2000.

2. At the time we invented the present invention, we were employed by NEC USA, INC. (hereinafter "NEC USA") and was employed at the C&C RESEARCH LABORATORIES at NEC USA.

3. Prior to October 19, 2000, the U.S. Filing Date of U.S. Patent No. 6,744,769, the invention as described and claimed in the above referenced application was completed at NEC USA, as evidenced by the following:

4. Prior to October 19, 2000, it was standard practice of employees of NEC USA to submit their inventions to the Legal Administrator at NEC USA on a CCRL Technical Report, after their inventions were approved by their supervisors, and for the invention to be assigned a Reference Number.

5. Prior to October 19, 2000, having earlier conceived the idea as set forth in the specification of the above referenced application, the present invention was formally submitted to the Legal Administrator at NEC USA in the form of a CCRL Technical Report. In the ordinary course of business and in due time, the present invention was assigned Reference Number CCRL 1106, and a CCRL Technical Report was prepared, as shown in Exhibit A, a copy of which is attached hereto.

6. Exhibit A discloses an Invention Disclosure Sheet with the CCRL Technical Report, which is an 18-page disclosure document prepared by us. Exhibit A includes subject matter that supports at least claims 11, 12 and 14-16. These claims of the present application are described at least in the following passages of our document in Exhibit A (further support may also be found elsewhere in Exhibit A):

Claim 11 – Pages 8-11

Claim 12 – Pages 1-4

Claim 14 – Pages 10-11

Claim 15 – Pages 2-6

Claim 16 – Page 4

7. The Invention Disclosure Sheet with the CCRL Technical Report was evaluated by Mr. Stephen B. Weinstein, Technical Manager and Area Manager, prior to October 19, 2000. Once the evaluation had been completed, the evaluation was reviewed and approved by Kojiro Watanabe, Vice President of NEC USA, prior to October 19, 2000, and then forwarded to us for final approval. We signed the approved papers on October 26, 2000.

8. At the time the subject matter of the present application was invented, it was common practice at NEC USA to have its provisional patent applications prepared and filed by persons not employed by NEC USA.

9. In the ordinary course of business and in due time, NEC USA sent a request to Sughrue Mion, PLLC, of Washington, DC, USA requesting filing of a provisional application and subsequent preparation and filing of a corresponding utility patent applications. The requests were sent from Mr. Yoshi Ryujin, Legal Administrator of NEC USA to Mr. Howard L. Bernstein of Sughrue Mion, PLLC on Friday, October 27, 2000. A copy of the above request is attached as Exhibit B.

10. In the ordinary course of business and in due course, Sughrue Mion, PLLC filed the provisional applications in the U.S. Patent Office and forwarded copies thereof to NEC USA on Monday, October 30, 2000. A copy of the letter from Sughrue Mion, PLLC is attached as Exhibit C.

11. In the ordinary course of business NEC USA reviewed and approved the draft applications prepared by Sughrue Mion, PLLC, and U.S. Application No. 09/897,495 was subsequently filed, properly claiming priority to the above-described provisional application.

DECLARATION UNDER 37 CFR §1.131
U.S. APPLICATION NO. 09/897,495

12. In view of the foregoing, it is clear that we, the named inventors of the above-captioned application, invented the subject matter of the claims prior to the October 19, 2000 U.S. filing date of U.S. Patent No. 6,744,769.

We hereby declare further that all statements made herein are of our own knowledge and are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Date: 12/29/05

Rauf Izmailov

Date: 12/21/05

Subir K. Biswas

Subir Biswas

Date: 12/29/05

Samrat Ganguly

Samrat Ganguly

EXHIBIT A

NEC-USA Invention Disclosure Sheet

ID# PTE 0041

TR#

Short Descriptive Title of Invention

Path Provisioning for service Level Agreements in Differentiated Service Networks

☑ 1. Plans to Publish: Is this an Internet draft or web-based submission? Yes □ No □

☑ a. Submission Date of Paper: October 31, 2000

□ b. Expected Publication Date: July, 2001

□ 2. No Plan to Publish

Related Technical Reports No.

TR 99-C030-4-5065-4

Search for prior arts: ☑ a. complete □ b. partial □ c. none

Key Words: routing. Diffserv

Inventor/s

1) Name of Inventor: Subir Biswas
Citizenship: India
Organization/University: CCRL
Tel(Dial in):
e-mail: YSKB15@YAHOO.COM
Home Address: 755 VILLAGE DRIVE
EDISON, NJ 08817
Signature: Subir K. Biswas
Date: 10/26/00

2) Name of Inventor: Samrat Ganguly
Citizenship: India
Organization/University: CCRL
Tel(Dial in):
e-mail:
Home Address: (a Summer Intern from Rutgers University)
Signature
Date:

3) Name of Inventor: Rauf Izmailov
Citizenship: USA
Organization/University: CCRL
Tel(Dial in):
e-mail: Rauf@ccrl.njnec.com
Home Address: 15 Elsive Drive
Plainsboro, NJ 08536
Signature
Date: 10/27/00

4) Name of Inventor
Citizenship:
Organization/University:
Tel(Dial in):
e-mail:
Home Address
Signature
Date:

5) Name of Inventor
Citizenship:
Organization/University:
Tel(Dial in):
e-mail:
Home Address
Signature
Date:

◆ **Evaluation of Invention (Technical and Area Managers only)**

1. **Patentability.**
☑ a. Yes □ b. No

2. **Related Business Division in NEC.**
☑ a. Name of the Division and the Manager
Division Name: 1st Lab, Development Labs Manager: Makiko Yoshida
□ b. Not related to Business Division in NEC

3. **Possibility of commercialization (Commercialization inside or outside NEC).**
□ a. Within 2 years □ b. Within 4 years ☑ c. Unknown

4. **Product group or technical area to which this invention's associated.**
QoS server in IP networks

5. **Schedule for implementation:**
A) implemented in the prototype
B) under consideration by the Division
C) scheduled to be implemented in the product
D) likely to be implemented within 5 years
[✓ E)] unknown

6. **Competitor of this technical area.**
Cisco

7. **Countries where patent would be applied other than USA**
☑ Japan □ Canada □ England □ France □ Germany □ China □ Korea
□ other (Country name)

8. **Remarks** (See attached Sheet)

Technical Manager
Date: [redacted]

9. **Remarks**

Area Manager
Date: [redacted]

◆ **Vice President use only**

1. **Execution**
☑ a. Send to Attorney □ b. Hold □ c. Keep as record

2. **Necessary for Prior Search by Attorney**
□ a. Yes ☑ b. No

3. **Patent Filing**
□ a. **Decision after the prior search.**
□ b. **Provisional Filing** **Due date:**
☑ c. **Determination of US filing**

1. Priority of filing in US
□ Express (Reason) ☑ within 3 months

2. Ownership
☑ NEC-USA only □ Joint with NEC
□ Joint with ( )

4. **Priority of Technical Importance**
□ a. Most important for NEC business ☑ b. Normal □ c. Less Important

5. **Remarks**

Signature:
Kojiro Watanabe
Date: [redacted]

CCRL No. ____________________

Remarks about patentability of Biswas-Ganguly-Izmailov work on "Path Provisioning for Service Level Agreements in Differentiated Services Networks"

This work describes high-performance algorithms for provisioning paths supporting DiffServ (Differentiated Services) aggregations of traffic requiring different quality of service (QoS) treatments. Making DiffServ work for QoS-sensitive traffic is a major objective of network operators. NEC will have a market edge for its QoS Manager equipment if this equipment is capable of better path provisioning than the QoS managers of other manufacturers.

Although efficient implementations of the proposed algorithms are not yet specified, I believe that they could be, realizing inventions that have at least the possibility of being significant contributions to competitive NEC products. Moreover, it is important for NEC to accumulate a protective portfolio of patents in the critical area of QoS in IP networks. For these reasons, despite the inventors' hesitation to propose this material for patentability study, I believe that a patentability study is justified.

Stephen B. Weinstein
Manager, Communications Technology Research
[redacted]

NEC-USA Invention Disclosure Sheet

DESCRIPTION OF THE INVENTION

a) The closest technology and disadvantages of the technology:
b) How is this problem solved by the invention (brief description of the new features of the invention).
c) Advantages of the invention over the closest technology.
d) Description of preferred embodiments.

a) Currently, the mechanisms for distributed path allocation are available. The disadvantage of this approach is its sub-optimality.

b) The proposed method permits to select paths in a pseudo-optimal way.

c) The path provisioning performance is improved.

d) Preferred embodiment is a server that collects information on traffic flows, compute paths and informs edge routers on its computations.

(USE INK) Provide a brief description of the invention and attach prints, reports, photos, etc. as available for full and complete understanding of the invention, including its operation and environment
(USE ADDITIONAL SHEETS AS REQUIRED.)

LIST ALL ATTACHMENTS

Add copy of the Publications and reports

- Publications of the closest technology
- Publications useful for understanding the invention

S. K. Biswas, S.Ganguly and R. Izmailov

Path Provisioning for Service Level Agreements in Differentiated Services Networks

Abstract:

We study the path provisioning as a mechanism to deliver Service Level Agreements in IP Differentiated Services networks. The problem of path provisioning is an NP-complete problem, so we propose and analyze (by simulations) several algorithms for solving the problem. As our simulations demonstrate, a centralized server consistently delivers a better performance than a distributed solution. We also show that the performance of one of the proposed algorithms, the greedy algorithm with backtracking, can be very close to the optimal one, while being computationally feasible.

Path Provisioning for Service Level Agreements in Differentiated Services Networks

S. K. Biswas, S.Ganguly[1] and R. Izmailov

NEC USA C&C Research Laboratories

1 Abstract

We study the path provisioning as a mechanism to deliver Service Level Agreements in IP Differentiated Services networks. The problem of path provisioning is an NP-complete problem, so we propose and analyze (by simulations) several algorithms for solving the problem. As our simulations demonstrate, a centralized server consistently delivers a better performance than a distributed solution. We also show that the performance of one of the proposed algorithms, the greedy algorithm with backtracking, can be very close to the optimal one, while being computationally feasible.

2 Characterization of IP Differentiated Services

IP Differentiated Service (Diffserv) has been evolving as an efficient and scalable technique for providing QoS within the Internet. The primary goal of Diffserv is to move away from per-flow signaling (considered to be unscalable because of its per-flow state storage in the core network) for achieving end-to-end QoS guarantees. The per-flow scalability problem is avoided in Diffserv by dealing with IP traffic as class-based aggregated flows as opposed to IntServ approach based on application level flows.

2.1 Diffserv Classes

In a Diffserv network, each IP packet belongs to one of a number of pre-defined classes and is given a service quality specified by its Diffserv class. A detailed description of the Diffserv architecture can be found in [1]; here we briefly describe only the components of the Diffserv model relevant to our study.

When an IP packet enters a Diffserv domain (a network area implementing Diffserv specifications), it is first classified at the edge and then marked with a Diffserv Code Point (DSCP) [2] that is used for identifying its traffic class. The DSCP is encoded at the COS/TOS field [2] in

[1] This author is at the Department of Computer Science, Rutgers University and was working in CCRL as a summer intern.

the IP packet header. Packet classification can be done based on layer-3, layer-4 or any other information that can be extracted from the packet. The exact classification process is outside of the scope of Diffserv definition and is left open for specific implementations. Once a packet is marked with a specific DSCP, the routers within the core handle the packet based on its Diffserv class. A router exercises class-specific queuing, scheduling and routing in order to satisfy the quality guarantees that are specified for the individual classes. Each Diffserv domain is free to define the scope of its Diffserv classes and mechanisms to provide the differentiated treatment. In other words, a packet can be classified and marked differently by the ingress routers of two different Diffserv domains. Neighboring Diffserv domains can enter into Service Level Agreement (SLA) [3]. These agreements could specify class-specific amount of traffic to be sent between the domains. Class-specific policing at the ingress and shaping at the egress is used for satisfying the SLAs.

Scheduling guidelines for Diffserv classes are provided by IETF by a concept called Per Hop Behavior (PHB) [4]. This document also specifies the PHBs for a number of DiffServ classes. IETF is also in the process of standardizing the usage of the DSCP for coding the Diffserv classes. Currently, five standard classes, namely, Expedited Forwarding (EF) [4], Assured Forwarding-1 (AF1), AF2, AF3 and AF4 [5] are specified. Within each AF class, three subclasses are defined based on their packet drop precedences. In addition to these standardized classes, an ISP can define non-standard Diffserv classes using locally-significant DSCP values.

2.2 End-to-End Service Characterization

Although the Diffserv standard specifies class-specific per-hop forwarding behaviors, it does not impose any specific end-to-end service characterization for the individual Diffserv classes. This is deliberately left up to carriers and vendors to customize service profiles and their mapping to individual Diffserv classes.

Diffserv Classes	Performance Bounds	Application
Expedited Forwarding	Very low delay/jitter Very low loss (10^{-9})	TDM Circuit Emulation
Assured Forwarding -1	Low delay/jitter Low loss (10^{-6})	CD quality audio Broadcast quality video
AF2	Low delay Low loss (10^{-4})	IP Telephony (voice and video)
AF3	Loss (10^{-3})	E-Commerce Web Trading
AF4	Loss (10^{-2})	FTP Telnet Non-critical Web Applications

Table 1: Requirements and application mapping for the standard Diffserv classes

Each Diffserv class within an ISP's domain is characterized by an end-to-end loss bound and an end-to-end delay bound. For instance, AF1 can be associated with CD-quality audio with a loss upper bound of 10^{-6}. Multiple AF classes are distinguished in terms of their individual loss and delay guarantees. The EF class can be characterized as one with arbitrarily small loss and queuing delay [4], which can be achieved by priority scheduling for EF packets. A possible mapping of several Internet applications to standard Diffserv classes is shown in Table 1.

Within an ISP domain (both Point of Presence and backbone), class-specific scheduling and routing are required for meeting the specified performance bounds of the Diffserv classes. At routers, specific scheduling mechanisms has to be in place, based on the class-specific PHBs. Expedited Forwarding, for instance, requires preemptive priority scheduling in order to secure its extremely low tolerance to queueing. On the other hand, AF classes can be processed using Weighted Fair Queuing (WFQ) [7] with appropriate scheduling weights. These scheduling mechanisms within a router are assumed to be non-adaptive. In other words, once the scheduling for a class on a router interface is set, it does not change with the varying traffic profile of the class.

3 Service Level Agreements

The most basic Service Level Agreements (SLA) specification [3] is the aggregated amount of traffic that would enter through an ingress router. For example, the SLA for aggregated traffic from domain-A to domain-B through ingress router I_1, in Figure 1, is specified as the rate λ. Domain-A estimates this rate from long-term traffic measurements and its own contracts with its other neighbor domains. In Figure 1, traffic flows only for this particular SLA are shown as solid arrows. All other flows are shown as dotted arrows. Note that the SLA parameters are used for policing at the ingress routers (e.g. router I_1) and shaping/conditioning at the egress routers (e.g. router E_3).

The SLA description for $E_3 \rightarrow I_1$ can be more specific where the portions of traffic λ, going to different egresses are also specified. The traffic from domain-*A* that arrives at domain-*B* through ingress I_1, is expected to go to the egress routers E_1 and E_2. The corresponding rates λ_1 and λ_2 (where $\lambda=\lambda_1+\lambda_2$) are specified in the SLA. As we shall explain later, this flow distribution information is extremely useful for Diffserv path provisioning and other traffic engineering purposes. However, if domain-*A* is not able to estimate this fine-grain flow distribution, then SLA contract will not have it. In that case, domain-*B* will have to estimate it from dynamic traffic measurement.

SLAs are specified for aggregated traffic and their duration of validity is generally much larger than the duration of the end-applications and their corresponding micro-flows. It is important to realize that an SLA specification may or may not capture the short-term rate variations that are important for path provisioning and traffic engineering. Short-term traffic rate variations have to be handled by other mechanisms.

Note that the class-specific resource partitioning and scheduling are not captured within the SLA specification. SLAs only describe the inter-domain traffic profiles. The usage of SLAs for intra-domain path provisioning and traffic engineering is decided solely by the local policies within a domain. The constraints, of course, are to satisfy the contractual QoS guarantees for the individual

SLAs. The next section is devoted to describing our intra-domain algorithms for path provisioning for satisfying the SLAs.

From the SLA information, each ingress router of a domain can compute the estimated volume of class-specific traffic from itself to all other egress routers in the same domain. This way, for every class, an $N \times N$ traffic matrix is created, where N is the number of edge routers in the domain. The (i,j)th element of the traffic matrix for a class represents the total bandwidth this class uses from ingress router i to the egress router j. For example, in Figure 1, for domain-B, the (1;1)th element is λ_1 , and the (1,2)th element is λ_2. We refer to this element as flow from i to j. Once the traffic matrices are constructed, the next task is to compute the provisioning routes for each non-zero element of those matrices. The computed paths are pinned down using either IP source routing or a layer-2 mechanism like MPLS.

For the EF class, the peak rate (λ in Figure 1) of SLA specification is used in the traffic matrices. For the AF class, however, the token bucket parameters are lumped into a single rate parameter that can be used for constructing the traffic matrix. The token bucket parameters may be used for computing the equivalent bandwidth of an SLA specification [5]. For example, the rate information for every AF SLA in Figure 1, is specified as $\{\lambda_p, \lambda_s, B\}$, where λ_p is the peak bucket rate, λ_m is the mean bucket rate and B is the maximum burst size). The interpretation of equivalent bandwidth is as follows. If an aggregated flow is allocated with its equivalent bandwidth and the flow complies with its SLA token bucket parameters, then the packets of that flow will be guaranteed the QoS specified by the flow. Since the equivalent bandwidth captures the loss requirements, the route computation algorithms ensure the loss bounds are automatically guaranteed. The delay bounds are not considered in this paper.

Note that the equivalent bandwidth computation is based on local policies and algorithms within the domain. We assume that the class-specific resource partitioning, scheduling and AF QoS requirements are uniform within the domain and known to all its routers. The discussion of the equivalent bandwidth computation algorithms is outside of the scope of this paper. Instead, we concentrate on provisioned path routing algorithms that can be used once a traffic matrix is constructed using a local algorithm for equivalent bandwidth computation.

4 Path Provisioning Problem for Differentiated Services Networks

In addition to PHBs, class-specific routing is used for achieving the end-to-end intra-domain QoS for the Diffserv classes. Packets with Diffserv marking can be forwarded using statically provisioned end-to-end paths. Class-specific forwarding paths with bandwidth reservation are provisioned for every source-destination pair within an ISP's domain. Paths are computed based on the static SLAs for individual Diffserv classes. Any significant change in the service level agreement causes re-computation of the provisioned paths. Since path provisioning relies on static SLAs, it is more likely to be used in the ISPs' core networks where multiple traffic flows produce less dynamic aggregated behavior and therefore, reasonably accurate SLA characterization is possible.

Routes for the provisioned paths can be computed centrally by a policy/QoS manager like the Bandwidth Broker in [6] or by the source nodes in a distributed manner. In the remaining part of the paper, we propose and analyze different algorithms for computing routes for provisioned Diffserv paths.

We partition the problem of path provisioning for Diffserv classes into multiple problems, each handling its own Diffserv class. The problems are then solved sequentially, starting form the most stringent class, EF, to the least demanding one, AF4. First, the provisioned paths for the EF traffic matrix are computed and pinned down. The amount of available bandwidth on the links used in these paths is correspondingly adjusted (subtracting the amounts reserved for EF from the available capacity of links). Then, the same procedure is repeated for other Diffserv classes (from AF1 to AF4). For each traffic class, we have the following formal problem.

Consider a directed graph $G = (N,E)$ with N nodes and E links. We associate two real numbers with each link e of the graph:

- C_e is the link capacity;
- B_e is the available bandwidth of the link, where $C_e \geq B_e$.

Each element (traffic flow) of the traffic matrix (SLA matrix) is a triplet (r,s,d), where

- s is an ingress node;
- d is an egress node;
- r is traffic rate from ingress node s to egress node d.

The triplets are referred to as $T(i)=(r_i, s_i, d_i)$, where $i=1,\ldots,K$ and K is the total number of triplets (non-zero elements of traffic matrix). Some triplets (flows) may be accepted and others rejected because there may not be enough bandwidth available to accept all. The accepted flows have to be routed while keeping in mind the following three criteria.

1. We denote by R the number of rejected triplets $T(j_1),\ldots,T(j_R)$; the remaining K–R triplets are accepted. The ratio $\overline{R} = R/K$ (we call it *flow blocking rate*) should be as low as possible.
2. We denote by V and W the total amounts of bandwidth of accepted flows and all flows, respectively:
$$V = \sum_{i \notin \{j_1,\ldots,j_R\}} r_i , \quad W = \sum_{i \in \{j_1,\ldots,j_R\}} r_i .$$
The ratio $\overline{V} = V/W$ (we call it *traffic acceptance rate*) should be as high as possible.
3. We denote by C the amount of network resources used to accommodate the K–R accepted flows:
$$C = \sum_{i \in \{j_1,\ldots,j_R\}} r_i h_i ,$$
where h_i is the number of hops in the path for the accepted triplet $T(i)$. The number C (we call it *hop-bandwidth product*) should be as low as possible.

The performance metrics $\overline{R}$, $\overline{V}$ and C are conflicting targets: since flows have different bandwidth requirements, there is usually a choice of accepting fewer "large" flows at the expense of more "small" flows or vice versa. Since accepted bandwidth directly translates to the revenue earned by the ISP, our first priority is to maximize the traffic acceptance rate $\overline{V}$ by deciding whch flows are accepted and which are rejected. The second priority is to route in a way to minimize the hop-bandwidth product C.

Note that splitting [11] of individual flows is an efficient way of balancing network loads. Although flow splitting may increase the total bandwidth of accepted traffic flows, ensuring packet ordering for individual micro-flows is difficult with splitting. It may require per-packet layer-3 lookup and hashing at the ingress routers. In this paper, we do not allow flow splitting; for each element in a traffic matrix only one ingress-to-egress path is chosen and subsequently pinned down.

We consider several approaches to the path-provisioning problem.

In the first approach, the i^{th} edge router first computes the traffic vector to all the other (N–1) edge routers. This vector corresponds to the i^{th} row of the traffic matrix (SLA information) of the domain. If the SLA information is not locally available, it may be necessary for a central SLA manager to download the relevant SLA specifications to the i^{th} edge router. Then, independently of other routers, each edge router computes and pins down the provisioning paths from itself to all other edge routers in the domain.

Since the ingress routers compute and pin down the paths independently, the result of this provisioning can be described as follows.

1. List all triplets $T(i)$ in an arbitrary sequence.
2. If the list of triplets is not exhausted, select the next triplet $T(i)$; else, stop.
3. For triplet $T(i)=(r_i, s_i, d_i)$, compute the shortest path p_i from s_i to d_i that satisfies the current bandwidth availability of the network: $B_e > r_i$ for for each link e in the path p_i. This is done by pruning all the links e that cannot support the traffic rate r_i (for these links, $B_e < r_i$) and computing the shortest path on the remaining subnetwork.
4. If such a path p_i exists, recompute B_e for each link e in the path p_i as $B_e := B_e - r_i$.
5. Go to step 2.

We refer to this approach as *naïve algorithm* and denote it as NA. We use this algorithm as a benchmark for other algorithms, in order to evaluate the benefits of having a QoS server handling path provisioning computations. The performance of NA approximates that of distributed path provisioning, where edge routers select paths for their traffic flows independently of each other and the bandwidth availability information is propagated to the routers by QOSPF protocol.

A QoS server can coordinate the order and manner of path selection. Problems of that nature have been addressed previously in the literature. There has been some work done on single path routing service in the context of telephone networks and virtual private network; the routing algorithms used in these networks depend on specific switching equipment provided by manufacturers. Lin and Wang [8] discuss the scenario with single path routing where cost to be minimized is the maximum link utilization factor. The authors cast this as a linear programming problem using Lagrangian Relaxation for obtaining suboptimal solutions. As pointed out in Frei and Faltings [9,10], the path-provisioning problem is a NP-complete and should be attacked by heuristic algorithms.

In this paper, we consider two heuristic algorithms. First of them is a greedy algorithm which we referred to as *iterated sorting* algorithm (denoted IS). In this algorithm, we sort the triplets in terms of the first field i.e., the rate. Thus we assume that the triplets $T(i)=(r_i, s_i, d_i)$ satisfy the condition $r_1 \geq r_2 \geq \ldots \geq r_K$, where K is the total number of triplets with non-zero rates (SLA entries). The sorted list is then handled in the same manner as in the NA algorithm.

In each step, the IS algorithm picks the largest (in terms of bandwidth required) flow from the list of non-rejected flows and attempts to fit it in the network. The algorithm may be suboptimal, as illustrated in Figure 2. All the links of the network in Figure 2 have available capacity of 10 units, and there are only two triplets in the SLA: (6,5,6) and (5,1,4). Provisioning the path (5,2,3,6) for the first triplet (6,5,6) blocks the second triplet, (5,1,4).

The blocking of the triplet (5,1,4) can be prevented by backtracking and reversing the sequence of paths to be provisioned: first, accommodate the triplet (5,1,4) with the path (1,2,3,4) and then the triplet (6,5,6) is routed along the path (5,7,8,9,6). This is the idea behind our second heuristic algorithm (greedy algorithm with backtracking), which we call *sequential path shifting* and refer to as SPS. To describe the algorithm formally, we use the following notations.

Given a graph G, we define for each triplet $T(i)=(r_i, s_i, d_i)$ two paths:

- Ideal shortest path: SPI(i) is the shortest path in G from s_i to d_i such that for all links e in the path, $C_e \geq r_i$. In other words, SPI(i) is the shortest path in the absence of bandwidth reservations of other triplets.
- Available shortest path: SPA(i) is the shortest path in G from s_i to d_i such that for all the links e in the path, $B_e \geq r_i$. In other words, SPA(i) is the shortest path in the presence of bandwidth reservations of other triplets.

For any path p carrying the bandwidth reservation r we define the following two operations: **Add** and **Del**:

- **Add**(G,p,r): for each link e in the path p, the available bandwidth B_e is decreased by r: $B_e := B_e - r$. In other words, **Add**(G,p,r) adjusts the available bandwidth in G in a way that reflects the reservation of bandwidth amount r along the path p.
- **Del**(G,p,r): for each link e in the path p, the available bandwidth B_e is increased by r: $B_e := B_e + r$. In other words, **Del**(G,p,r) adjusts the available bandwidth in G in a way that reflects the release of bandwidth amount r along the path p.

As in the IS algorithm, the SPS algorithm uses the list of triplets sorted in terms of the first field *i.e.*, the rate. The triplets are then sequentially selected according to this order. For each triplet, a decision is made whether to accept it or not. This decision may involve changing the paths for already accepted triplets. In order to incorporate this backtracking, we expand the triplets to quadruplets by adding an extra bit b. The bit indicates whether the provisioned path for the flow can be altered by subsequent flows. In the beginning, the bit is set as TRUE in for all quadruplets (the paths for all flows can be altered).

For $i = 1, \ldots, K$, the SPS algorithm sequentially tries to find a path from the ingress to egress for the ith quadruplet. In step i, we consider the quadruplet $T(i)=(r_i, s_i, d_i, b_i)$ for which the path has to be computed. Let H_i and H_i^* denote the number of hops in SPI(i) and SPA(i), respectively; if SPA(i) is not defined (this happens when there is not enough bandwidth in the network to route the

i^{th} flow), we set H_i^* to infinity. The difference between H_i^* and H_i is the number of hops by which SPA(i) exceeds the optimal path SPI(i). Therefore, we define the sub-optimality cost $W(i)$ as $r_i(H_i^* - H_i)$. This cost represents the amount of additional bandwidth used by the i^{th} flow when compared with the optimal path SPI(i). If $W(i)$ is zero (the available path uses the same amount of bandwidth as the ideal one), the path SPA(i) is accepted for $T(i)$ and the step i is completed.

Otherwise, if $W(i) > 0$, the path SPI(i) cannot be used to accommodate the i^{th} flow since for at least one of the links e in SPI(i), its available bandwidth B_e is smaller than the flow rate r_i. We denote the set of all such links by Q, where $Q = \{e : e \in \text{SPI}(i); B_e < r_i\}$. If there were no other flows in the network, all links of SPI(i) would be able to accommodate flow rate r_i. However, because of paths already provisioned during the previous i-1 steps, some of the links of SPI(i) (namely, those belonging to Q) do not have the available bandwidth necessary to support the flow rate r_i.

We denote by M the subset of those already accepted (during the previous i-1 steps) quadruplets $T(1),\ldots, T(i-1)$ for which the following two conditions hold:

- The bit r_j of quadruplet is TRUE. Therefore, the path SPA(j) can be altered.
- All links e in Q belong to the path SPA(j): $Q \subset \text{SPA}(j)$. Therefore, if the bandwidth reservation for r_j of the quadruplet T(j) for its path SPA(j) is removed, the available bandwidth at each link e in Q increases by r_j. Since the i^{th} flow requires bandwidth reservation of $r_i \leq r_j$, this increase is sufficient for accommodating the i^{th} flow using its path SPI(i).

If the set M is empty and SPA(i) is defined, we accept it as the path for the i^{th} flow (the path is sub-optimal, but there is no single path we can remove to accommodate it); otherwise, we reject the i^{th} flow.

If M is not empty, any of its elements can be used to accommodate the i^{th} flow in the following way. For each quadruplet $T(j)$ of the set M, its path SPA(j) contains all the links in Q. Therefore, removing the path SPA(j) from the network by releasing the bandwidth reserved for T(j), permits the ith flow to be accommodated with on the path SPI(i). The removed quadruplet $T(j)$ now has to be routed again, and its new altered route may be longer than the one scheduled originally. For each quadruplet $T(j)$ of this set ($j \in M$), we define the shifting cost S_j as $r_j(L_j^* - L_j)$ where L_j and L_j^* are calculated in the following way.

- L_j is the number of hops in the current path $P(j)$ for the quadruplet $T(j)$.
- L_j^* is the number of hops in the path $P^*(j)$ calculated in G^* for $T(j)$, where G^* is obtained from G by first performing the operation Del($G,P(j),r_j$) and then the operation Add(G,SPI(i),r_i) (these operations reflect the result of deleting the bandwidth provisioning for T(j) and adding the bandwidth provisioning for $T(i)$). If path $P^*(j)$ does not exist, we set L_j^* to infinity.

Denote by S_{min} the minimum of all the shifting costs S_j, achieved for some j=m. Depending on what is larger, S_{min} or $W(i)$, one of following two actions is taken.

1. If $S_{min} > W(i)$, the shifting cost of the previously processed quadruplet $T(m)$ exceeds the sub-optimality cost of quadruplet $T(i)$. Therefore, we accept SPA(i) as the path for $T(i)$ and do not change the path for $T(m)$.

2. If $S_{\min} \leq W(i)$, the shifting cost of the previously processed quadruplet $T(m)$ is smaller than the sub-optimality cost of quadruplet $T(i)$. Therefore, we route the quadruplet $T(m)$ on the path $P^{*}(m)$ and reset SPA(m)=$P^{*}(m)$ and we route the quadruplet $T(i)$ on the path SPI(i) and reset SPA(i)=SPI(i). Also, if the shift of $T(m)$ resulted in a path longer than L_m (which was the number of hops in the path $P(j)$ for the quadruplet $T(m)$ before the shifting), we change the bit b_i of the quadruple $T(i)$ to FALSE. This is done to prevent the shifting of SPA(i) by subsequent flows (i+1, ..., N) and to simplify the algorithm (if SPA(i) can be shifted by a subsequent flow $T(k)$, the altered path for the quadruplet $T(m)$ can be changed again (as well as the paths that were shifted by quadruplet $T(m)$ etc.), which leads to state-space explosion).

5 - Performance of Path Provisioning Algorithms

In the previous section, we described three path provisioning algorithms: NA, IS and SPS. We analyze their performance for two networks under various loading conditions.

We start with the network as shown in Figure 3: it represents the physical topology of the IP Backbone. The Backbone network consists of 12 nodes, and we assume that every link can carry 10 units of bandwidth. We selected the following three sources and three destinations.

- **Sources:** Seattle (1), San Francisco (2) and Los Angeles (3).
- **Destinations:** Cambridge (8), New York (9) and Washington, DC (10).

Each source generated three flows to all three destinations, which created nine flows in total. The traffic rate of each flow was randomly distributed on the interval (0,10ρ), where ρ is a scale parameter that is varied form 0 to 1. The average traffic rate of each flow is thus equal to ρ/2. We tested nine different values of ρ (namely, ρ=0.1, ρ=0.2,..., ρ=0.9) and, for each value of ρ, we ran all three algorithms (NA, IS and SPS) 15 times and computed the following.

- Average traffic acceptance rate ***R***, defined as the average accepted traffic volume *R* normalized by the total traffic volume (which is 9ρ/2 in this particular example).
- Average flow rejection rate ***V***, defined as the average number *V* of rejected traffic flows normalized by the total number of traffic flows (which is 9 in this particular example).
- Average hop-bandwidth product ***C***, defined as the average hop-bandwidth product of *accepted* flows.

For this series of experiments, we also used a fourth algorithm, the *brute force* one (referred to as BF). In this algorithm, for each of the nine source-destination pairs, we enumerated all the paths that consist of no more than four hops (the restriction of all paths being no longer than four hops was introduced only to limit the explosion of the solution space; in fact, in some rare situations, we observed path-provisioning solutions involving longer paths). As a result, there are $F = 7\times11\times11\times12\times15\times17\times9\times10\times11 = 2{,}565{,}901{,}800$ different ways to select nine paths (four hops or less) for all nine flows. If none of *F* ways provides a solution (solution is defined as a set of routes maximizing ***R*** and, for the sets with the same ***R***, minimizing ***C***), one of the nine flows has to be blocked (there are nine ways to do it), and the remaining eight flows can be routed to their destinations by reduced sets of eight paths. For instance, if the flow from source 2 to destination 10 is dropped, then none of the 17 paths from for this source-destination pair is used, and the remaining eight flows can be routed in *F*/17=150,935,400 ways. Overall, eight flows can be routed in

$$F/7+F/11+F/11+F/12+F/15+F/17+F/9+F/10+F/11=2{,}143{,}859{,}850$$

ways. If none of these ways provides a solution, two flows have to be dropped (there are 72 ways to do it), and the remaining seven flows can be routed to their destinations by reduced sets of seven paths, and so on.

While the performance of NA provides a lower bound of performance, the performance of BF algorithm provides a useful upper bound. Using this bound, we can judge how close the proposed algorithms are to the optimal one. BF is not meant to be used as an actual path provisioning algorithm in realistic scenarios; it takes about 12 hours on a Pentium-120 Mhz to solve the problem for just one set of flows. We used BF algorithm in the series of experiments to calculate the same performance metrics as for NA, IS and SPS.

The results of our experiments are shown on Figures 4–6. For small values of ρ, we observe that there is hardly any difference in all three performance metrics. In other words, for small loads, all algorithms perform about the same. However, as the average flow requirement reaches 0.3 (and the overall load on the system increases), differences in performance emerge.

In terms of average traffic acceptance rate (which is our primary target performance metrics) there is a visible difference between the decentralized NA algorithm, on one side, and the centralized IS, SPS and BF algorithms, on the other side (Figure 4). The relative difference among latter algorithms is relatively smaller. Within the set of these algorithms, SPS performs better than IS, and BF performs better than SPS. The difference between BF and SPS is small, which indicates that the performance of SPS is close to that of the optimal BF algorithm.

In terms of flow blocking rate (which is our secondary performance metric), the relationship between NA and other algorithms (IS, SPS and BF) is reversed: NA accepts, on average, more flows than other algorithms (Figure 5). The reason for this reversal is the tradeoff between traffic acceptance rate and flow blocking rate: IS, SPS and BF accommodate more traffic volume by accepting fewer flows with larger bandwidth requirements.

Finally, in terms of hop-bandwidth product (which is our third performance metric), there is hardly any difference between the algorithms (Figure 6).

We also experimented with larger traffic matrices consisting of 16 flows (Figures 7–9) and 25 flows (Figure 10–12). In all these experiments, only NA, IS and SPS were tested, without BF (even with 16 flows, it would take too much time to compute the optimal allocation). As in the case of 9 flows, the algorithms have the same performance for lower loads; as the average flow bandwidth increases, differences in performance emerge. These differences exhibit the same patterns as in the case of 9 flows. Namely, IS and SPS deliver higher traffic acceptance rate (with SPS slightly outperforming IS) than that of NA, while the flow blocking rate of IS and SPS are higher than that of NA. Also, the hop-bandwidth product for IS and SPS are higher than that of NA. The reason for this is the sub-optimality of the paths used to accommodate the additional traffic flows.

Besides the IP Backbone network, we experimented with the Kyoto University ATM network (Figure 13). The network consists of 5 core nodes fully connected by 10 logical links of1244 Mbps capacity (a logical link consists of two parallel physical links of 622Mbps each). The core nodes are connected to the gateway nodes (switches) of 8 domains (departments) with links of 622 Mbps. Inside each domain, there are 8 to 10 (normal) nodes, interconnected with 155Mbps links. Some of these normal nodes have direct connections to the 5 core nodes via 622 Mbps links, bypassing the corresponding gateway node. These normal nodes (with direct connections to the core nodes) are called bypass nodes, and each domain has exactly one bypass node. Traffic was sent from each intra-domain node to every other intra-domain node (there are 64 intra-domain nodes in Kyoto University network. The traffic intensity depends on the relationship between the source and destination nodes; 50% of the traffic is intra-domain traffic, whereas the other 50% of the traffic goes uniformly to the other seven domains. Since the Kyoto University network is not symmetrical, this arrangement leads to dissimilar traffic loading on different links of the network.

The experiments with the resulting 64×64 traffic matrix are shown in Figures 14–16. The relative performance of the algorithm follows the same pattern observed in our experiments with IP Backbone. In particular, the SPS algorithm delivers the best traffic acceptance rate, followed closely by IS.

As our experiments demonstrate, there is some value in handling path provisioning problem by a centralized QoS server. Our simulations also demonstrated that the performance of one of our algorithms (greedy algorithm with backtracking) can be very close to the optimal one, while being computationally feasible.

6 Summary and Conclusions

We considered the path provisioning problem in Diffserv networks. We proposed and analyzed several algorithms solving the problem. We identified the greedy algorithm with backtracking algorithms as being consistently the best one for the most important performance metric. In future work, we plan to study the issue of path provisioning for less precisely defined SLAs, where exact breakdown of outgoing traffic is not known. We also plan to address the option of multiple paths used for the same traffic flow.

7 References

1. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, *An Architecture for Differentiated Services,* IETF Request for Comments 2475, October 1998.

2. K. Nichols, S. Blake, F. Baker and D. Black, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,* IETF Request for Comments 2474, December 1998.

3. M. Biegi, R. Jennings, S. Rao and D. Verma, *Supporting Service Level Agreements using Differentiated Services,* IETF Internet Draft <draft-verma-diffserv-ntimplem-00.txt>, November 1998.

4. V. Jacobson, K. Nichols and K. Poduri, *An Expedited Forwarding PHB,* IETF Request for Comments 2598, June 1999.

5. J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, *Assured Forwarding PHB Group,* IETF Request for Comments 2597, June 1999.

6. K.Nichols, V. Jacobson, L. Zhang, *A Two-bit Differentiated Services Architecture for the Internet,* IETF Request for Comments 2638, July 1999.

7. D.Stephens, H.Zhang, "*Implementing Distributed Packet Fair Queueing in a Scalable Switch Architecture*". Proceedings of IEEE INFOCOM'98.

8. F.Lin and J.Wang, "*A Minimax Utilization Routing Algorithm in Networks with Single-Path Routing,*" *IEEE Globecom'93.* 1993.

9. "*Resource allocation in networks using constraint satisfaction,*" Iconomic Systems White Paper, http://www.iconomic.com.

10. C. Frei and B.Faltings, "*Abstraction and Constraint Satisfaction Techniques for Planning Bandwidth Allocation*", Proceedings of IEEE Infocom, 2000.

11. S. Biswas, R. Izmailov and B.Sengupta, "*Connection Splitting: An Efficient Way of Reducing Call Blocking in ATM,*" in Proceedings of GLOBECOM'98, Sydney, Australia, November 8--12, 1998, pp. 2412--2418.

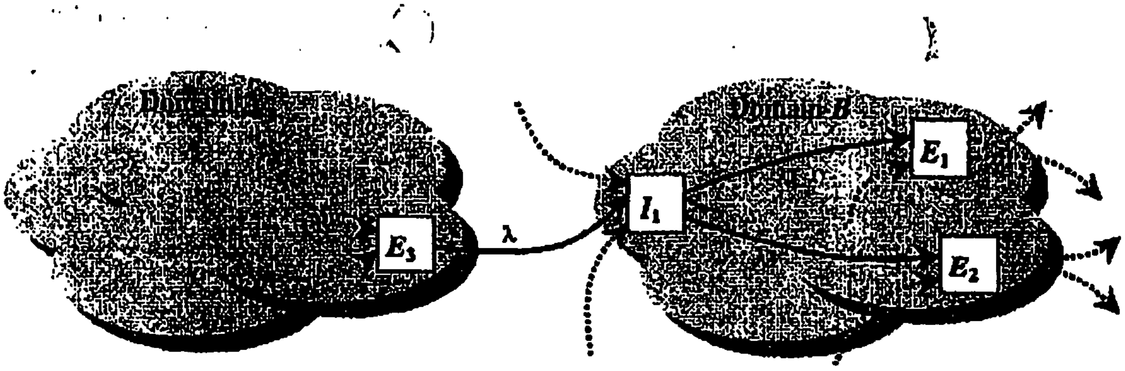

Figure 1: Inter-domain Service Level Agreement.

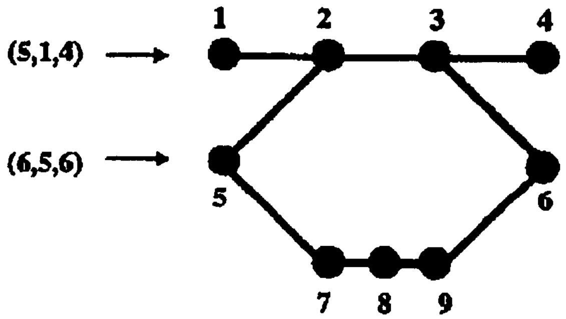

Figure 2: Sub-optimality of the greedy algorithm IS.

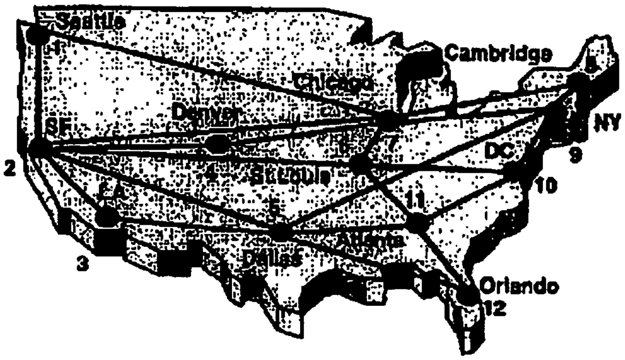

Figure 3: Physical topology of IP Backbone.

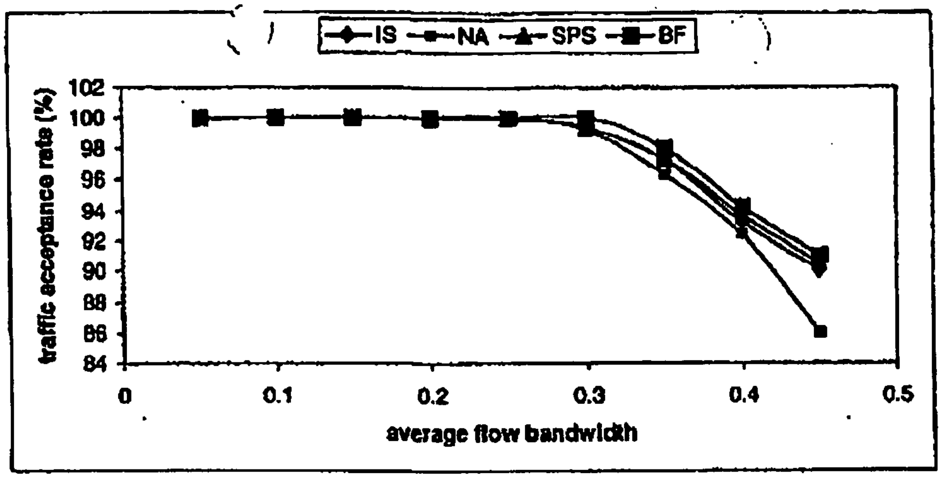

Figure 4: IP Backbone: traffic acceptance rate for 9 flows.

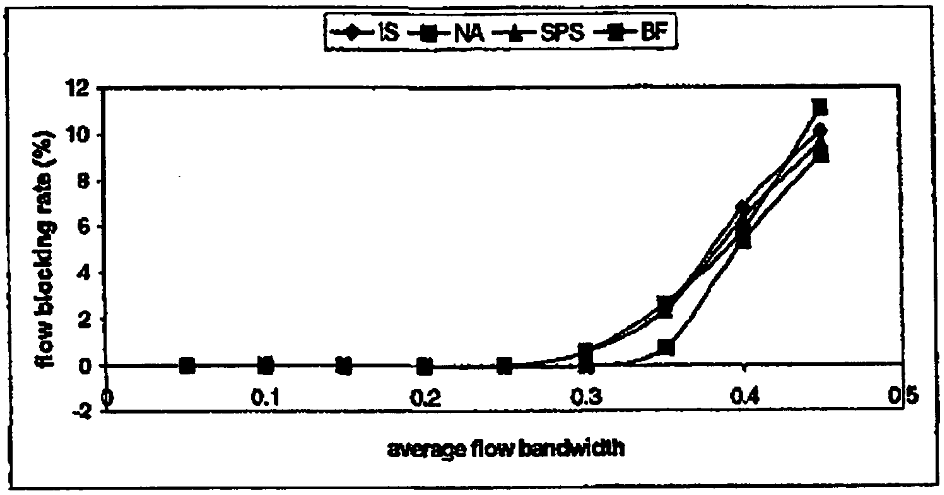

Figure 5: IP Backbone: flow blocking rate for 9 flows.

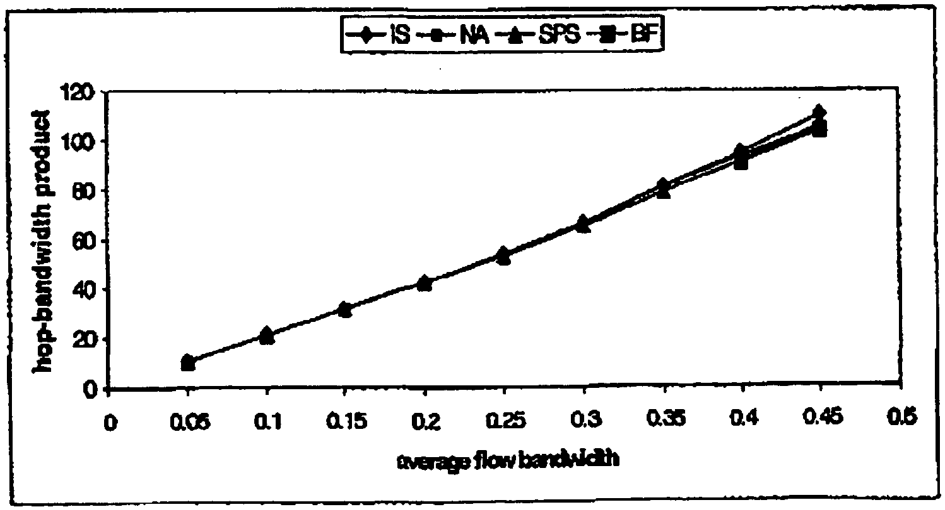

Figure 6: IP Backbone: hop-bandwidth product for 9 flows.

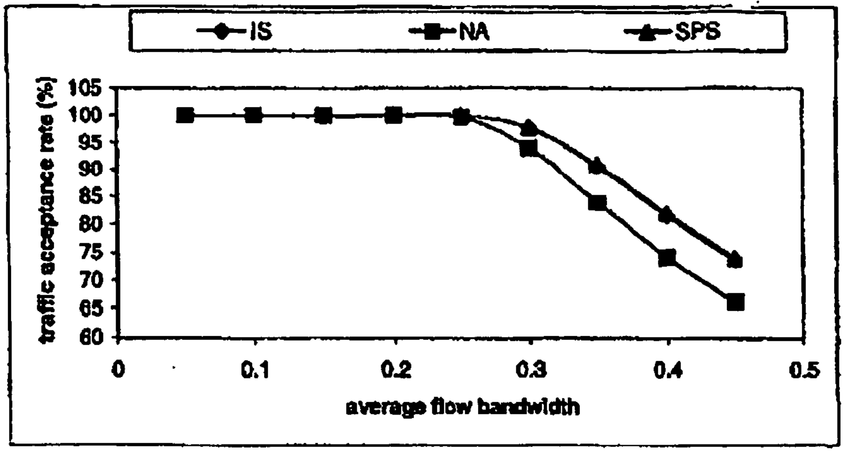

Figure 7: IP Backbone: traffic acceptance rate for 16 flows.

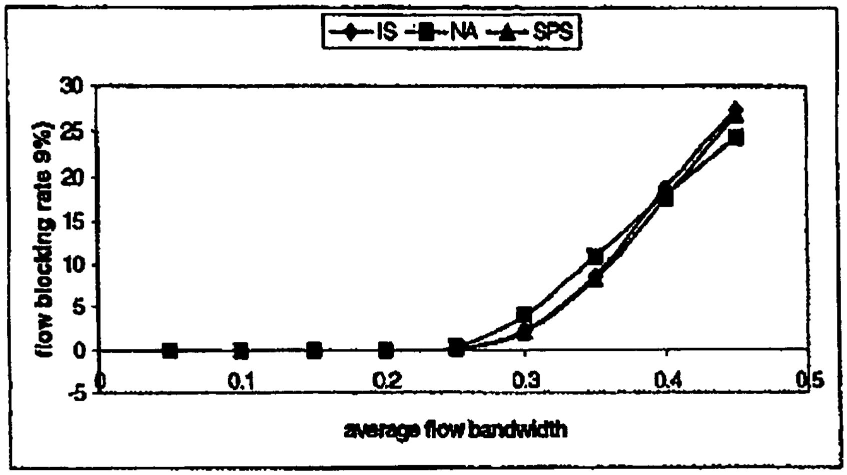

Figure 8: IP Backbone: flow blocking rate for 16 flows.

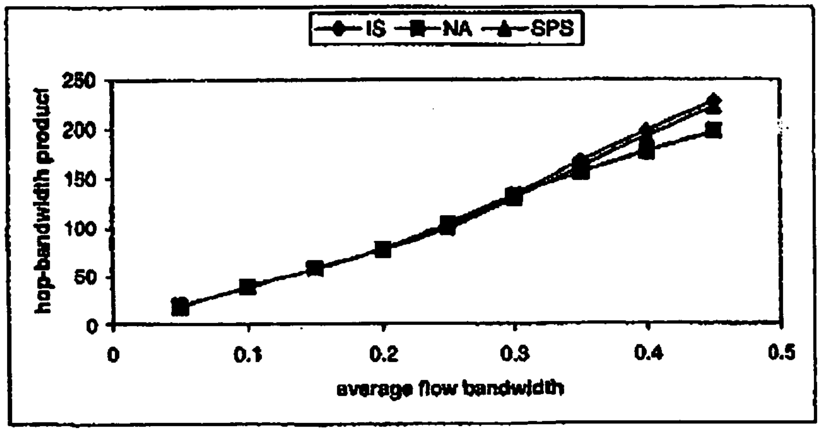

Figure 9: IP Backbone: hop-bandwidth product for 16 flows.

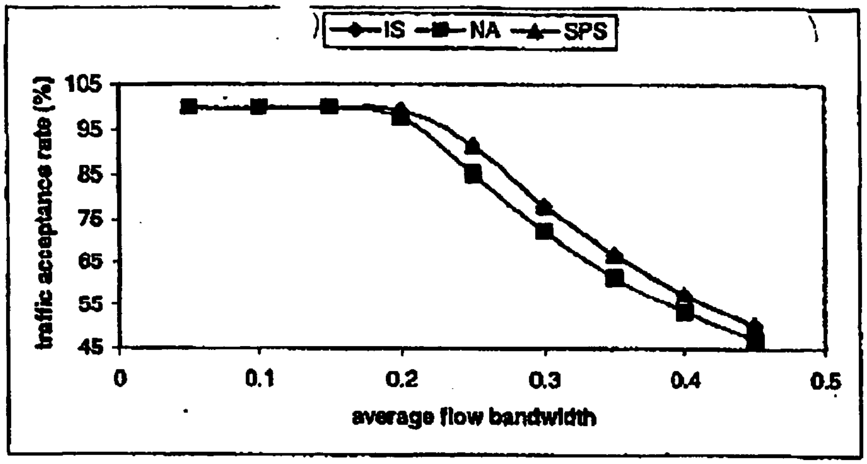

Figure 10: IP Backbone: traffic acceptance rate for 25 flows.

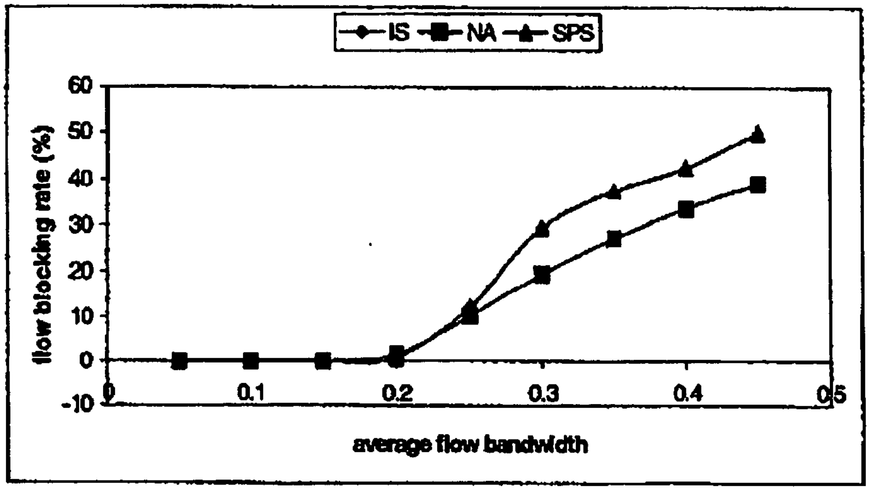

Figure 11: IP Backbone: flow blocking rate for 25 flows.

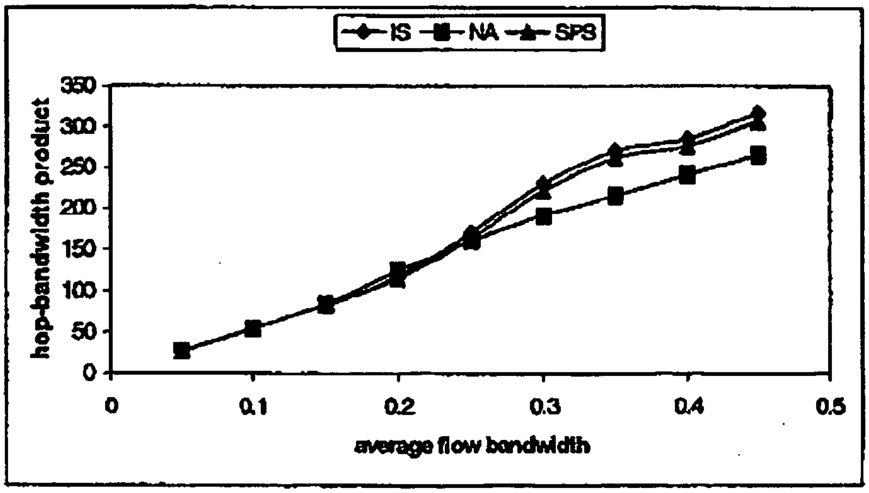

Figure 12: IP Backbone: hop-bandwidth product for 25 flows.

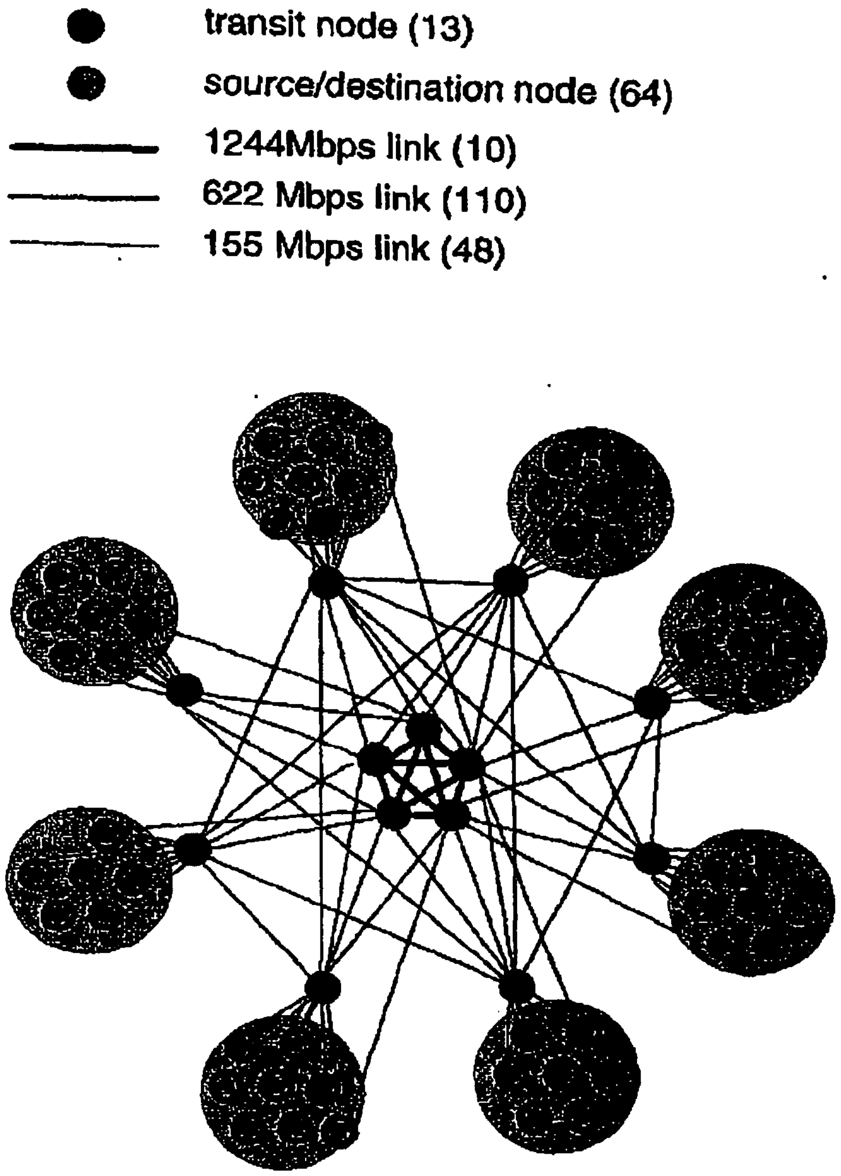

Figure 13: Physical topology of Kyoto University network.

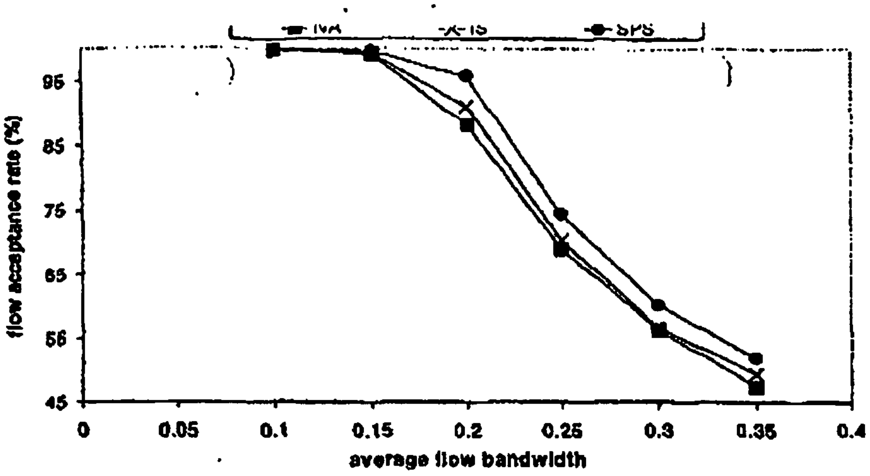

Figure 14: Kyoto University: traffic acceptance rate.

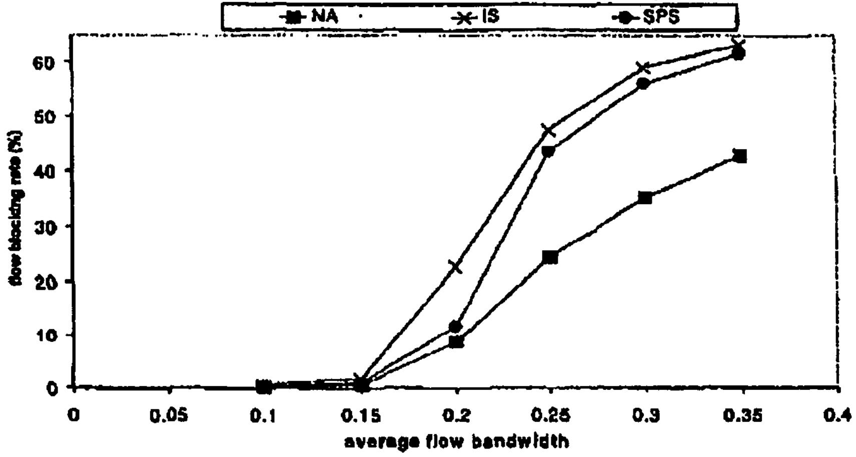

Figure 15: Kyoto University: flow blocking rate.

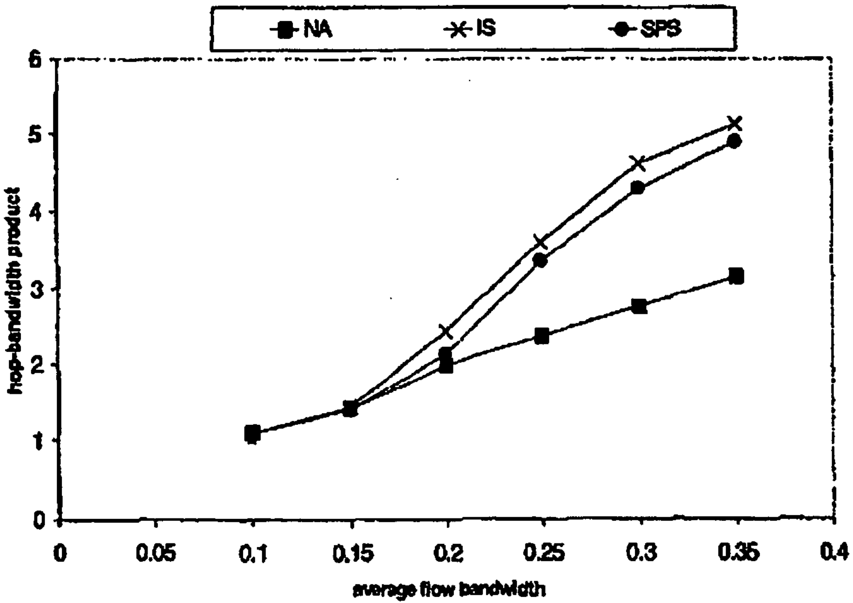

Figure 16: Kyoto University: hop-bandwidth product

EXHIBIT B

NEC

NEC USA, Inc.
C&C Research Laboratories
4 Independence Way
Princeton, New Jersey 08540
Tel. 609-951-2999
Fax 609-951-2499

VIA FAX

October 27, 2000

Mr. Howard L. Bernstein
Sughrue, Mion, Sinn, Macpeak & Seas
2100 Pennsylvania Avenue, NW, Suite 800
Washington, D.C. 20037-3202

Re: A request of New U.S. Patent Application
Title: Path Provisioning for Service Level Agreements in Differentiated Service Networks
Inventor: Rauf Izmailov, Subir Biswas, Samrat Granguly
Our Ref. No: CCRL 1106

Dear Mr. Bernstein:

This is a request to ask you to file a new patent application based on the attached invention disclosure form. A provisional application must be filed by October 30,2000, as the submission date of the paper is October 31, 2000. And we would like you to file the regular Patent Application to U.S. P.T.O. within three months.

Attached is a copy of invention disclosure form and CCRL Technical Report, which title is "Path Provisioning for Service Level Agreements in Differentiated Service Networks" by Rauf Izmailov, Subir Biswas and Samrat Granguly.

Your inventor contact is Rauf Izmailov and he can be contacted at 609-951-2454; email: rauf@ccrl.nj.nec.com.

If you have any questions, please contact Mr. Rauf Izmailov or me.

Very truly yours,

Yoshi Ryujin
Legal Administrator

CC: Laurent Desclos
J. Kanai
M .Kondo

EXHIBIT C

LAW OFFICES

SUGHRUE, MION, ZINN, MACPEAK & SEAS, PLLC

2100 PENNSYLVANIA AVENUE, N.W.
WASHINGTON, DC 20037-3213
TELEPHONE (202) 293-7060
FACSIMILE (202) 293-7860
www.sughrue.com

Howard L. Bernstein
Direct Dial (202) 663-7937
Email: hbernstein@sughrue.com

October 30, 2000

Mr. Yoshi Ryujin
NEC USA, INC.
C & C Research Laboratories
4 Independence Way
Princeton, NJ 08540

Re: Inventors Subir BISWAS, Samrat GRANGULY, and Rauf IZMAILOV
"PATH PROVISIONING FOR SERVICE LEVEL AGREEMENTS IN DIFFERENTIATED SERVICES NETWORKS"
Assignee: NEC USA, INC.
Filing of Provisional Application
Your Ref: CCRL 1106
Our Ref: P7870
Due Date: January 30, 2000 (filing of regular patent application)

Dear Yoshi:

Thank you for your facsimile of October 27, 2000 with Invention Disclosure Form and Technical Report for the above-identified New U.S. application. In accordance with your instructions, we have prepared and filed a Provisional Application in the Patent Office today. I enclose two copies of the Provisional Application as filed.

Thank you for referring this matter to us.

Very truly yours,

Howard L. Bernstein

HLB/mrp
Enclosures

cc: Junko Kanai, NEC CRL (w/o Enclosures)